A New Nonlinear Solution Method for Phase Change Problems

D.A. Knoll, D.B. Kothe, and B. Lally

Los Alamos National Laboratory

M.S. D413

Los Alamos, New Mexico 87545

Running head: New Phase Change Algorithm

Abstract

We present a new nonlinear algorithm for the efficient and accurate solution of isothermal and nonisothermal phase change problems. The method correctly evolves latent heat release in isothermal and nonisothermal phase change, and more importantly, it provides a means for the efficient and accurate coupling between temperature and concentration fields in multi-species nonisothermal phase change. The method rigorously conserves energy both globally and locally. Newton-like super-linear convergence is achieved in the global nonlinear iteration, without the complexity of forming or inverting the Jacobian matrix. This "Jacobian-free" method is a combination of an outer Newton-based iteration and an inner conjugate gradient-like (Krylov) iteration. The effects of the Jacobian are probed only through approximate matrix-vector products required in the conjugate gradient-like iteration. The methodology behind the Jacobian-free Newton-Krylov solution method is given in detail. We demonstrate the properties of this method which allow the formulation of an implicit solution algorithm having enthalpy as the dependent variable. The performance of the method is demonstrated on phase change problems for a pure material undergoing isothermal solidification and a binary eutectic alloy undergoing nonisothermal solidification.

Nomenclature

C_p = heat capacity

C_l = solute concentration in liquid

C_s = solute concentration in solid

h = phase enthalpy

H = enthalpy

$\mathbf{J}$ = Jacobian matrix

k = partition coefficient

L = latent heat of fusion

$\mathbf{M}$ = preconditioning matrix

m_l = liquidus slope

St = Stefan number

T = temperature

t = time

T_{eut} = eutectic temperature

T_f = fusion temperature

T_l = liquidus temperature

T_m = melting temperature

$\mathbf{u}$ = velocity

$\mathbf{v}$ = Krylov vector

x = position

$\mathbf{y}$ = preconditioned Krylov vector

Nomenclature (cont.)

Greek

α = thermal diffusivity

ϵ = small perturbation

ϵ_s = solid volume fraction

κ = thermal conductivity

Subscripts

l = liquid

s = solid

1 Introduction

Numerical simulation is playing an increasingly important role in support of industrial casting processes. The goal of simulations is to accurately capture the solidification dynamics in pure materials, and in multicomponent alloys. To achieve this goal a numerical algorithm must accurately evolve the latent heat in an isothermal solidification process, and it must also accurately couple the temperature and concentration fields in the nonisothermal solidification of multicomponent alloys.

Implicit solution algorithms are often preferred to avoid undesirable numerical stability (time step) restrictions. The most common form of the energy equation for implicit methods uses temperature as a dependent variable [1, 2, 3]. This form of the energy equation includes a latent heat source term which is not a single valued function of temperature for isothermal solidification. Several strategies have been developed for integrating this source term implicitly. Those strategies which attempt to conserve energy are forced to fix locations on the grid which are undergoing isothermal solidification [1, 2]. At these locations, where the energy equation is actually being solved for enthalpy, a local change of dependent variable has occurred.

We present a fully implicit solution method for the enthalpy form of the energy equation. The front evolves in a completely self-consistent fashion and there is no local switching of dependent variables. Enthalpy is the dependent variable in our nonlinear iterations, and temperature is represented as a local function of enthalpy and solid volume fraction in a pure material; in an alloy, temperature is represented as a local function of enthalpy, solid volume

fraction, the phase diagram, and a local scale transport model.

For multicomponent problems with flow, our algorithm assumes that species advection is modeled in a time explicit fashion, consistent with the assumption in Ref. [2]. However, the basic numerical method presented here could be used as a general flow solver [4, 5, 6]. One could solve the coupled system for a solidification problem with flow with this technique, which is currently under investigation.

We view our algorithm as a departure from existing fixed grid implicit methods. It is an implicit method with enthalpy as the dependent variable, a global Newton iteration is performed without forming or inverting the Jacobian matrix, and the solidification front is captured rather than being tracked. With this in mind, our primary motivation is to clearly describe the details of the method. Detailed performance comparisons with existing algorithms will be the focus of a subsequent study.

In the following sections we discuss the issue of an enthalpy versus a temperature form of the energy equation. We present the structure of our discrete implicit equation for enthalpy, and the associated temperature function. This is illustrated for a pure material Stefan problem and for a binary eutectic alloy with a simple local scale model. Details of our Newton iteration are then presented , specifically we discuss how the Newton iteration is accomplished without forming or inverting the Jacobian. We conclude with some typical performance results for a pure material Stefan problem and for a binary eutectic alloy.

2 Temperature or Enthalpy Formulation

We start with the definition of a two-phase mixture energy equation appropriate for materials undergoing solid-liquid phase change. The subscript s denotes the solid phase and the subscript l denotes the liquid phase. In enthalpy form, the evolution of energy of the solid-liquid mixture is given by:

$$\frac{\partial[\rho H]}{\partial t} + \nabla \cdot (\rho_l h_l \mathbf{u}_l) - \nabla \cdot (\kappa \nabla T) = 0, \tag{1}$$

where the mixture density is,

$$[\rho] = \epsilon_s \rho_s + (1 - \epsilon_s)\rho_l, \tag{2}$$

and the mixture volumetric enthalpy is,

$$[\rho H] = \epsilon_s \rho_s h_s + (1 - \epsilon_s)\rho_l h_l. \tag{3}$$

The phase enthalpies are,

$$h_s = C_{p,s} T, \tag{4}$$

$$h_l = C_{p,l} T + L. \tag{5}$$

Here ρ is density, $\mathbf{u}$ is velocity, T is temperature, ϵ_s is the solid volume fraction, κ is the thermal conductivity, C_p is the heat capacity, and L is the latent heat of fusion. For simplicity, and ease of communication, we present our method with the following assumptions:

1. one spatial dimension;

2. no flow in the liquid ($\mathbf{u}_l = 0$);

3. constant thermodynamic properties (ρ, κ, C_p, L) within phases;

4. and equal thermodynamic properties between phases ($\rho_s = \rho_l$, , etc.).

These assumptions are not necessary for our algorithm, but rather simplify the discussion that follows.

These assumptions result in a simplified mixture enthalpy equation,

$$\rho \frac{\partial H}{\partial t} - \kappa \frac{\partial}{\partial x}(\frac{\partial T}{\partial x}) = 0, \tag{6}$$

where $H = C_p T + (1-\epsilon_s)L$. Upon substitution into Eq.(6) and rearrangement we can obtain the temperature form of the energy equation.

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial}{\partial x}(\frac{\partial T}{\partial x}) = -\frac{L}{C_p}\frac{\partial(1-\epsilon_s)}{\partial t}, \tag{7}$$

where $\alpha = \kappa/(\rho C_p)$. This form of the energy equation is most often employed in implicit fixed grid methods for phase change problems [1, 2, 3]. Various strategies have been developed for integrating the source term implicitly while maintaining energy conservation in the phase change region [1, 2, 3]. The main difficulty with this approach is evident in Fig. 1, where a pure material enthalpy-temperature relation is depicted. It is clear that there exists a range of possible values for H at T_m(the melting temperature), i.e. there is not a unique $H(T_m)$. The infinite slope at this point is also observed in nonisothermal solidification for alloys with a eutectic point. The temperature version of the energy equation at the front is therefore ill-posed because $H(T_m)$ is not single-valued. At the front, T is constant and ϵ_s is a function

of enthalpy through the relationship $H - C_p T_m = (1 - \epsilon_s)L$. This non-uniqueness is avoided in standard implicit methods by fixing the temperature T at T_m, whereby the energy equation is solved for ϵ_s with the time derivative of temperature set to zero. This can be shown to be equivalent to solving for H at the front. Substituting $H - C_p T_m = (1 - \epsilon_s)L$ into Eq.(7) (with $\frac{\partial T}{\partial t} = 0$), the energy equation at the front becomes,

$$\alpha \frac{\partial}{\partial x}(\frac{\partial T}{\partial x}) = \frac{1}{C_p} \frac{\partial (H - C_p T_m)}{\partial t}, \tag{8}$$

or upon rearrangement,

$$\kappa \frac{\partial}{\partial x}(\frac{\partial T}{\partial x}) = \rho \frac{\partial (H)}{\partial t}, \tag{9}$$

This represents a local exchange in dependent variables at the front from T to H, and requires one to fix the front position over a nonlinear iteration within a time step. This artificial "fixing" of the front position may restrict time step size for nonlinear convergence and accuracy. However, this approach is certainly to be preferred to the apparent heat capacity method in terms of accuracy and energy conservation [1]. It is insightful to note that Eq.(9) is precisely the form of the energy equation that we started with, Eq.(6).

By employing enthalpy as the dependent variable on the entire grid we need to evaluate T as a function of H, which is depicted in Fig. 2 for a pure material. Here we can clearly see that temperature is a unique, single-valued function of enthalpy.

3 The Discrete Implicit Enthalpy Equation

In this section a discrete implicit solution to the enthalpy form of the energy equation is formulated. This results in a nonlinear system of algebraic equations. We demonstrate how to incorporate temperature as a function of enthalpy for both pure materials and alloys. A clear understanding of our nonlinear algebraic system is essential for describing the unique way in which we solve our nonlinear problem using a Newton iteration without forming the Jacobian.

Assuming that we can express temperature as a function of enthalpy, $T = \mathcal{T}(H)$, we write the enthalpy form of the energy equation as;

$$\frac{\partial H}{\partial t} - \frac{\kappa}{\rho}\frac{\partial}{\partial x}\left[\frac{\partial \mathcal{T}(H)}{\partial x}\right] = 0. \tag{10}$$

Using a simple first-order in time (backward Euler), second-order in space discretization on a uniform grid, a discrete version of this equation can be written;

$$\frac{H_i^{n+1} - H_i^n}{\Delta t} - \frac{\kappa}{\rho \Delta x^2}[\mathcal{T}(H_{i+1}^{n+1}) - 2\mathcal{T}(H_i^{n+1}) + \mathcal{T}(H_{i-1}^{n+1})] = 0, \tag{11}$$

where i is the grid index, n is the time step index, Δt is the time step, and Δx is the grid spacing. It is straightforward to construct a second-order in time, second-order in space, method as well. One could consider the Crank-Nicolson method,

$$\frac{H_i^{n+1} - H_i^n}{\Delta t} - \frac{\kappa}{\rho \Delta x^2}[\mathcal{T}(H_{i+1}^{n+1/2}) - 2\mathcal{T}(H_i^{n+1/2}) + \mathcal{T}(H_{i-1}^{n+1/2})] = 0, \tag{12}$$

where $H_i^{n+1/2} = (H_i^n + H_i^{n+1})/2$, or an Adams-Bashford three-level method,

$$\frac{3}{2}\frac{H_i^{n+1} - H_i^n}{\Delta t} - \frac{1}{2}\frac{H_i^n - H_i^{n-1}}{\Delta t}$$
$$-\frac{\kappa}{\rho \Delta x^2}[\mathcal{T}(H_{i+1}^{n+1}) - 2\mathcal{T}(H_i^{n+1}) + \mathcal{T}(H_{i-1}^{n+1})] = 0. \qquad (13)$$

To close the problem statement the function $\mathcal{T}(H)$ must be defined. For a pure material with a melting temperature T_m, the function is defined as;

$$\mathcal{T}(H) = \begin{cases} H/C_p, & H < C_p T_m; \\ T_m, & C_p T_m \leq H \leq C_p T_m + L; \\ (H - L)/C_p, & H > C_p T_m + L; \end{cases}$$

which is a piece-wise linear relationship between H and T (Fig. 2). This problem has been solved with a standard Newton method where the Jacobian is formed [7, 8]. It is our goal to construct a Newton-based method which easily extends to nonisothermal solidification problems by sidestepping the need to form the true Jacobian.

Moving to a more complex multicomponent system, such as a binary eutectic alloy, the function $\mathcal{T}(H)$ becomes more complex. Assuming a straight liquidus slope on the phase diagram we have the following temperature-concentration relationship,

$$T = T_f + m_l C_l, \qquad (14)$$

where T_f is the fusion temperature of the pure solvent, m_l is the slope of the liquidus line, and C_l is the liquid solute concentration. Assuming thermodynamic equilibrium holds at the solid-liquid interface results in the following relationship for the solute concentration in the solid at the interface,

$$C_s = kC_l, \tag{15}$$

where k is the partition coefficient. In nonisothermal solidification the region $0 < \epsilon_s < 1$ is referred to as the "mushy zone." In the mushy zone a local scale model is used to account for solute diffusion in the solid [9, 10, 2]. This local scale model defines the solute liquid concentration as a function of solid volume fraction,

$$C_l = G(\epsilon_s). \tag{16}$$

Here a solution to the energy equation must be found which simultaneously satisfies Eqs.(14) and (16), and $H = C_pT + (1 - \epsilon_s)L$.

In this study we will use simple local scale models which assume either complete solute diffusion in the liquid and no solute diffusion in the solid (the Scheil assumption), or complete solute diffusion in both the liquid and the solid (the lever rule). For a binary alloy (neglecting flow in the liquid), these assumptions result in the following temperature-solid volume fraction relationships.

$$\epsilon_s = 1.0 - \left(\frac{T_f - T_l}{T_f - T}\right)^{-(1-k)}, \tag{17}$$

for the Scheil assumption, and

$$\epsilon_s = \left(1.0 - \left(\frac{T_f - T}{T_f - T_l}\right)\right)\left(\frac{1}{1-k}\right), \tag{18}$$

for the lever rule.

The temperature function for a eutectic can be expressed as,

$$\mathcal{T}(H) = \begin{cases} H/C_p, & H < C_p T_{eut}; \\ T_{eut}, & C_p T_{eut} \leq H \leq C_p T_{eut} + (1-\epsilon_{s,eut})L; \\ (H - (1-\epsilon_s)L)/C_p, & C_p T_{eut} + (1-\epsilon_{s,eut})L \leq H \leq C_p T_l + L; \\ (H-L)/C_p, & H > C_p T_l + L; \end{cases}$$

where T_{eut} is the eutectic temperature.

For alloys, the additional complexity, as compared to pure materials, is that in the mushy zone $(C_p T_{eut} + (1-\epsilon_{s,eut})L \leq H \leq C_p T_l + L)$ where one must simultaneously satisfy the energy balance, the phase diagram, and a local scale model. In this region a small nonlinear system must be solved to extract T as a function of H. If the Schiel or lever assumptions hold, the phase diagram and local scale model can be combined to give ϵ_s as a function of T as in Eq.(17) or Eq.(18). Then the nonlinear system which needs to be solved to extract T as a function of H in the mushy zone consists of two equations, either Eq.(17) or Eq.(18) along with $H = C_p T + (1-\epsilon_s)L$. For example, if the Schiel model is being used, then the solution to the two equation nonlinear system,

$$
\begin{aligned}
C_p T + (1-\epsilon_s)L - H &= 0, \\
\epsilon_s + \left(\frac{T_f - T_l}{T_f - T}\right)^{-(1-k)} - 1.0 &= 0,
\end{aligned}
\tag{19}
$$

must be found at each finite volume in the mushy zone. This is done with a standard Newton iteration (typically in three to five iterations). For a binary alloy possessing a more complicated phase diagram and/or local scale model, solutions to a system of three nonlinear equations must be found, Eqs.(14) and (16) along with $H = C_p T + (1-\epsilon_s)L$. Again, this small nonlinear system is solved only for cells in the mushy zone. For an N component alloy a system of N+1 equations must be solved, Eq.(14), N-1 equations like Eq.(16) along with $H = C_p T + (1-\epsilon_s)L$.

The nonlinear functions resulting from the discretized energy equation (not $\mathcal{T}(H)$) play such an important role in describing the algorithm, we define them so as to avoid any confusion. The nonlinear functions are the discretized equations at each grid cell. The functions for energy at cell i, F_i, are

$$
F_i = \frac{H_i^{n+1} - H_i^n}{\Delta t} - \frac{\kappa}{\rho \Delta x^2}\left[\mathcal{T}(H_{i+1}^{n+1}) - 2\mathcal{T}(H_i^{n+1}) + \mathcal{T}(H_{i-1}^{n+1})\right]. \tag{20}
$$

In the next section we demonstrate that this function only needs to be evaluated with the current solution, $\mathbf{H}^n$, and a perturbed solution, $\mathbf{H}^n + \epsilon\mathbf{v}$ (where $\mathbf{v}$ is a general vector), to perform a Newton iteration. To evaluate F_i we need to evaluate $\mathcal{T}(H_{i-1})$, $\mathcal{T}(H_i)$ and $\mathcal{T}(H_{i+1})$. For those control volumes which reside in the mushy zone, a small nonlinear system needs to be solved. This

local nonlinear iteration is inside our global outer Newton iteration. Again, the local evaluation of F_i in the mushy zone for a multicomponent system requires the solution to a small nonlinear system to extract T as a function of H. The L_2 norm of the nonlinear function, F_i, is used to declare convergence of the nonlinear iteration within a time step..

4 Jacobian-Free Newton-Krylov Method

In this section we provide a detailed description of our nonlinear iterative method. The discussion of Newton's method is standard. We will give some description of a Krylov based linear iterative method in order to elucidate the Jacobian-free aspect of the proposed method. We will not give a detailed description of Krylov methods, for which we would recommend the following texts [11, 12]. The nonlinear iteration scheme is an inexact, matrix-free Newton-Krylov method. By *inexact* Newton we mean that the convergence tolerance of the linear solver is proportional to the current nonlinear residual. By *matrix-free* Newton-Krylov we mean that the required Jacobian-vector product in each linear (Krylov) iteration is replaced by a finite difference approximation to the true Jacobian-vector product [13].

Newton's method requires the solution of the linear system

$$\mathbf{J}^k \delta \mathbf{H}^k = -\mathbf{F}(\mathbf{H}^k), \qquad \mathbf{H}^{k+1} = \mathbf{H}^k + \delta \mathbf{H}^k; \tag{21}$$

where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{F}(\mathbf{H})$ is the nonlinear system of equations, $\mathbf{H}$ is the state vector, and k is the nonlinear iteration index. For our one-

dimensional problem, discretized into N equations and N unknowns, we have

$$\mathbf{F}(\mathbf{H}) = \{F_1, F_2, ..., F_i, ..., F_N\}, \tag{22}$$

and

$$\mathbf{H} = \{H_1, H_2, ..., H_i..., H_N\}, \tag{23}$$

where i is the one-dimensional grid index. In vector notation, the $(i,j)^{th}$ element of the Jacobian matrix is

$$J_{i,j} = \frac{\partial F_i(\mathbf{H})}{\partial H_j}. \tag{24}$$

Forming each element of $\mathbf{J}$ requires taking analytic or discrete derivatives of the system of equations with respect to H. This can be both difficult and time consuming for nonisothermal solidification problems and for problems with thermodynamic data in tabular form.

4.1 Matrix-free Approximation

In this study the Generalized Minimal RESidual (GMRES) [14] algorithm is used to solve the linear system of equations given by Eq.(21). GMRES (or any other Krylov method such as Conjugate Gradients) defines and initial linear residual, $\mathbf{r_0}$ given an initial guess, $\delta\mathbf{H_0}$ (typically zero),

$$\mathbf{r_0} = -\mathbf{F}(\mathbf{H}) - \mathbf{J}\delta\mathbf{H_0}. \tag{25}$$

Note that the nonlinear iteration index, k, has been dropped. This is because the GMRES iteration is performed at a fixed k. l is the linear iteration index. The l^{th} GMRES iteration minimizes $\parallel \mathbf{J}\delta\mathbf{H}_l + \mathbf{F}(\mathbf{H}) \parallel_2$ with a least squares

approach. $\delta\mathbf{H}_l$ is constructed from a linear combination of the Krylov vectors, $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, ..., (\mathbf{J})^{l-1}\mathbf{r}_0\}$, obtained during the previous $l-1$ GMRES iterations. This linear combination of Krylov vectors can be written as,

$$\delta\mathbf{H}_l = \delta\mathbf{H}_0 + \sum_{j=0}^{l-1} \beta_j (\mathbf{J})^j \mathbf{r}_0, \tag{26}$$

where evaluating the scalars β_j is part of the GMRES iteration.

Upon examining Eq.(26) we see that GMRES requires the action of the Jacobian only in the form of matrix-vector products, which may be approximated by [13];

$$\mathbf{J}\mathbf{v} \approx [\mathbf{F}(\mathbf{H} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{H})] \; / \; \epsilon, \tag{27}$$

where $\mathbf{v}$ is a general Krylov vector (i.e. one of $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, ..., (\mathbf{J})^{l-1}\mathbf{r}_0\}$), and ϵ is a small perturbation.

Equation (27) is simply a first order Taylor series expansion approximation to the Jacobian, $\mathbf{J}$, times a vector, $\mathbf{v}$. For illustration consider the two coupled nonlinear equations $F_1(H_1, H_2) = 0$, $F_2(H_1, H_2) = 0$. The Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial F_1}{\partial H_1} & \dfrac{\partial F_1}{\partial H_2} \\ \\ \dfrac{\partial F_2}{\partial H_1} & \dfrac{\partial F_2}{\partial H_2} \end{bmatrix}.$$

Our method never forms this matrix, we instead form a vector which approximates this matrix multiplied by a vector. Working backwards from Eq.(27), we have;

$$\frac{\mathbf{F}(\mathbf{H}+\epsilon\mathbf{v})-\mathbf{F}(\mathbf{H})}{\epsilon} = \begin{pmatrix} \frac{F_1(H_1+\epsilon v_1, H_2+\epsilon v_2)-F_1(H_1,H_2)}{\epsilon} \\ \\ \frac{F_2(H_1+\epsilon v_1, H_2+\epsilon v_2)-F_2(H_1,H_2)}{\epsilon} \end{pmatrix}.$$

Approximating $\mathbf{F}(\mathbf{H}+\epsilon\mathbf{v})$ with a first order Taylor series expansion about $\mathbf{H}$, we have;

$$\frac{\mathbf{F}(\mathbf{H}+\epsilon\mathbf{v})-\mathbf{F}(\mathbf{H})}{\epsilon} \approx \begin{pmatrix} \frac{F_1(H_1,H_2)+\epsilon v_1\frac{\partial F_1}{\partial H_1}+\epsilon v_2\frac{\partial F_1}{\partial H_2}-F_1(H_1,H_2)}{\epsilon} \\ \\ \frac{F_2(H_1,H_2)+\epsilon v_1\frac{\partial F_2}{\partial H_1}+\epsilon v_2\frac{\partial F_2}{\partial H_2}-F_2(H_1,H_2)}{\epsilon} \end{pmatrix},$$

which simplifies to;

$$\begin{pmatrix} v_1\frac{\partial F_1}{\partial H_1}+v_2\frac{\partial F_1}{\partial H_2} \\ \\ v_1\frac{\partial F_2}{\partial H_1}+v_2\frac{\partial F_2}{\partial H_2} \end{pmatrix} = \mathbf{J}\mathbf{v}.$$

This matrix-free approach has many unique capabilities. Namely, Newton-like nonlinear convergence without *forming* or *inverting* the true Jacobian. In practice we do form a matrix for preconditioning purposes, hence our algorithm is not entirely matrix-free. However, since the matrix we form is far simpler than the true Jacobian of the problem, our algorithm is Jacobian-free.

To complete the description of this technique we provide a prescription for evaluating the scalar perturbation. In this study ϵ is given by,

$$\epsilon = \frac{1}{N||\mathbf{v}||_2} \sum_{m=1}^{N} b|H_m|, \tag{28}$$

where N is the linear system dimension and b is a constant whose magnitude is within a few orders of magnitude of the square root of machine roundoff ($b = 10^{-5}$ for this study).

4.2 Preconditioning

Typically one uses a simple iterative method as a preconditioner to GMRES. The purpose of preconditioning is to efficiently cluster the eigenvalues of the iteration matrix, which in turn will reduce the required number of GMRES iterations. For example, for right preconditioning, The system that is solved is

$$(\mathbf{J}\mathbf{M}^{-1})(\mathbf{M}\delta\mathbf{H}) = -\mathbf{F}(\mathbf{H}), \tag{29}$$

where $\mathbf{M}$ symbolically represents the preconditioning matrix and $\mathbf{M}^{-1}$ the inverse of the preconditioning matrix. The matrix $\mathbf{M}$ is chosen to be an approximation to $\mathbf{J}$. In practice, this inverse is only approximately realized through some standard iterative method, and one may think of it more as $\tilde{\mathbf{M}}^{-1}$. The right preconditioned analog to Eq.(27) is;

$$\mathbf{J}\tilde{\mathbf{M}}^{-1}\mathbf{v} \approx [\mathbf{F}(\mathbf{H} + \epsilon\tilde{\mathbf{M}}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{H})] \; / \; \epsilon. \tag{30}$$

Solutions to Eq. (30) proceed in two steps;

1. Solve (iteratively, and not to convergence) $\mathbf{M}\mathbf{y} = \mathbf{v}$ for $\mathbf{y}$

2. Perform $\mathbf{Jy} \approx [\mathbf{F}(\mathbf{H} + \epsilon\mathbf{y}) - \mathbf{F}(\mathbf{H})] \; / \; \epsilon$,

In our Jacobian-free Newton-Krylov method only the matrix $\mathbf{M}$ is formed and only the matrix $\mathbf{M}$ is iteratively inverted. There are two choices to be made here;

1. What linearization should be used to form $\mathbf{M}$?

2. What linear iterative method should be used to solve

   $\mathbf{My} = \mathbf{v}$?

We have already defined $\mathbf{J}$ as the matrix resulting from the Newton linearization of our nonlinear system. Our stated goal is to avoid forming this matrix, but to maintain Newton-like nonlinear convergence. The Jacobian for the constant property Stefan problem is actually quite easy to form analytically and has been done in Ref. [8]. For our one-dimensional problem this is a tridiagonal matrix, with the following entries on each row;

$$\begin{aligned} J_{i,i-1} &= \frac{\partial F_i}{\partial H_{i-1}} = -\frac{\kappa}{\rho \Delta x^2} \mathcal{T}'(H_{i-1}), \\ J_{i,i} &= \frac{\partial F_i}{\partial H_i} = \frac{1}{\Delta t} + 2\frac{\kappa}{\rho \Delta x^2} \mathcal{T}'(H_i), \\ J_{i,i+1} &= \frac{\partial F_i}{\partial H_{i+1}} = -\frac{\kappa}{\rho \Delta x^2} \mathcal{T}'(H_{i+1}). \end{aligned} \tag{31}$$

Here, $\mathcal{T}'$ is the derivative (slope in Fig. 2) of the temperature function for the Stefan problem. It is straightforward to see that,

$$\mathcal{T}'(H) = \begin{cases} 1/C_p, & H < C_p T_m; \\ 0, & C_p T_m \leq H \leq C_p T_m + L; \\ 1/C_p, & H > C_p T_m + L; \end{cases}$$

We will use a constant value, $\mathcal{T}'(H) = 1/C_p$, for all values of enthalpy when evaluating our preconditioning matrix $\mathbf{M}$. Thus each row of our preconditioning matrix is simply

$$\begin{aligned} M_{i,i-1} &= -\frac{\kappa}{\rho C_p \Delta x^2}, \\ M_{i,i} &= \frac{1}{\Delta t} + 2\frac{\kappa}{\rho C_p \Delta x^2}, \\ M_{i,i+1} &= -\frac{\kappa}{\rho C_p \Delta x^2}. \end{aligned} \tag{32}$$

This is not a significant change from the exact Jacobian for the Stefan problem, but it is a significant change from the exact Jacobian matrix which would result from a multicomponent nonisothermal solidification problem. It can be envisioned that computing accurate Jacobian elements in the mushy zone would be quite complicated and costly as a result of the local nonlinear systems which need to be solved. To use a standard Newton method one would have to evaluate

$$\mathcal{T}'(H) = (1 + \frac{\partial \epsilon_s}{\partial H} L)/C_p, \tag{33}$$

in the mushy zone. Our algorithm avoids this complexity by using the matrix-free Jacobian-vector product approximation, and using a much simpler linearization to form the preconditioning matrix, $\mathbf{M}$.

An inexact Newton convergence tolerance is used on the preconditioned GMRES method. This means that the level to which the linear problem is solved is proportional to the level of convergence of the nonlinear problem. This can be expressed as,

$$\| \mathbf{J}^k \delta \mathbf{H}^k + \mathbf{F}(\mathbf{H}^k) \|_2 < \gamma \| \mathbf{F}(\mathbf{H}^k) \|_2 . \tag{34}$$

For this study $\gamma = 1.0 \times 10^{-2}$ is used. Use of an inexact Newton tolerance will prevent us from seeing the theoretically promised quadratic convergence in the nonlinear iteration. However, as will be shown, super-linear convergence is obtained. Additionally, we use a damped Newton update, $\mathbf{H}^{k+1} = \mathbf{H}^k + d\delta\mathbf{H}^k$, where d is a scalar chosen such that the maximum change in enthalpy does not exceed ten percent in any given Newton iteration.

To summarize, within each time step, an inexact Newton-GMRES iteration is used to solve the nonlinear system arising from the implicit discretization of an enthalpy based energy equation. This inexact Newton-GMRES iteration achieves Newton-like nonlinear convergence solely through a special matrix-vector multiply routine in GMRES. The true Jacobian of the system is never formed and never inverted. For completeness, and improved communication, we express our implicit algorithm in template form. For a detailed GMRES template consult reference [11].

Time Step Loop (n is time step index)

$\mathbf{H}^{n+1} = \mathbf{H}^n$ (initial guess)

Form preconditioning matrix, $\mathbf{M}$ (Eq.(32)).

while (not converged) **do** (Newton Loop)

(k is nonlinear iteration index)

If $k = 1$ form $\mathbf{F}(\mathbf{H}^k)$

while (not converged) **do** (GMRES Loop)

(l is linear iteration index)

solving $(\mathbf{J^k M^{-1}})(\mathbf{M}\delta\mathbf{H}^k) = -\mathbf{F}(\mathbf{H^k})$ for $\delta\mathbf{H}^k$

Initial Krylov vector, $\mathbf{v}_0 = -\mathbf{F}(\mathbf{H^1})$ (Eq.(25))

(build $\mathbf{v_l} = \mathbf{J^k M^{-1} v_{l-1}}$, l^{th} Krylov vector)

preconditioning operation, iteratively solve $\mathbf{My} = \mathbf{v}_{l-1} \rightarrow \mathbf{y}$

(fixed number of Symmetric Gauss-Seidel iterations)

matvec operation $\Rightarrow \mathbf{J^k y} \approx [\mathbf{F}(\mathbf{H^k} + \epsilon\mathbf{y}) - \mathbf{F}(\mathbf{H^k})] \; / \; \epsilon$

Complete l^{th} GMRES iteration, compute β_j's (Eq.(26))

Linear Convergence ? (Eq.(34))

end (GMRES Loop)

Evaluate Newton damping scalar, d

$\mathbf{H}^{k+1} = \mathbf{H}^k + d\delta\mathbf{H}^k$

Evaluate $\mathbf{F}(\mathbf{H}^{k+1})$, Nonlinear Convergence ?

end (Newton Loop)

$\mathbf{H}^{n+1} = \mathbf{H}^{k+1}$

time = time + Δt

end (Time Step Loop)

5 Algorithm Performance

5.1 Pure Material Solidification

For the first model problem we consider a 1-D pure material isothermal solidification problem, in the region $0 \leq x \leq 4$, with $C_p = \rho = \kappa = 1.0$. The melting temperature, T_m is equal to 1.0 and the initial conditions for temperature are 2.0. A grid of 50 uniform finite volumes is used. At time equal to zero, the left boundary is set to a temperature of zero, and solidification proceeds. To compare against the analytic solutions available for the infinite domain Stefan problem would require a time dependent boundary condition on the right boundary of our finite domain. We instead choose to maintain the right boundary at $T = 2.0$ and use a solution at very small time step as a base solution. Simulations are performed for $L = 10, 1$, and 0.1, which represent Stefan numbers (St) of 0.1, 1.0 and 10 respectively. Effort for these problems is measured as total linear solves and average number of Newton iterations per time step. Accuracy is measured in two ways against a base solution at a small time step. The first measure is front position, X_f, where $T = T_m$. In this measure, if a finite volume has the temperature $T = T_m$ then we take the center of that volume as X_f. If no finite volume has the temperature $T = T_m$ then linear interpolation is used to define X_f between the two bounding finite volume centers. The second measure is a global L_2 norm measure given by,

$$L_2 \equiv \sqrt{\sum_{i=1}^{N} (\frac{T_i - T_i^{base}}{T_m})^2} \tag{35}$$

In all simulations presented the inexact Newton tolerance from Eq.(34) is $\gamma = 1.0 \times 10^{-2}$ and the L_2 norm of the nonlinear residual is required to drop below 1.0×10^{-5}. Tables 1, 2, and 3 display performance data for $St = 0.1,\ 1.0$ and 10.0 respectively. All simulations are performed with the first-order accurate time step. For all Stefan numbers the base solution is computed with a second-order accurate time step and $\Delta t = 1.0 \times 10^{-4}$. In all simulations a time step "ramp" is used, whereby the initial time step is ten percent of the final time step, and the first eight time steps are equivalent to two final time steps. The first column gives the final time step used and the factor by which this time step exceeds the explicit stability limit of the diffusion operator ($\Delta t_{exp} = \frac{\Delta x^2}{\alpha 2}$). The second column displays the total number of time steps required to reach the final simulation time, the third column is the average number of Newton iterations per time step, and the fourth column is the total number of linear systems which are solved (one per Newton iteration). Columns five and six are the accuracy measures. It can be seen for all Stefan numbers that the number of Newton iterations per time step scales only weakly with time step size. Our solution method can therefore use large time steps and require fewer total linear solves. The accuracy of these solutions also does not seem to be a strong function of time step size. This is most likely a result of the strong nonlinear convergence within a time step, and the fact that the front is captured naturally by our algorithm. Observing the sixth column one can see the characteristic first-order time step convergence. Figures 3, 4, and 5 exhibit enthalpy profiles for a large time step and a small time step from the three Stefan number problems in the region $0 \leq x \leq 2$. In all cases the front position is captured

accurately over a large range of time steps.

Figure 6 displays the nonlinear convergence history at two different time steps for $St = 0.1$ and $\Delta t = 0.4$. The nonlinear residual is defined as $\sqrt{\sum_{i=1}^{N}(F_i(H^n))^2}$. At time step 26 the front did not cross into a new cell, while in time step 24 the front did move across a cell.

5.2 Binary Eutectic Solidification

As a second model problem we consider a 1-D binary eutectic problem, which has been adapted from Ref. [2], in the region $0 \leq x \leq 0.4$. This problem is done in dimensional form with $C_p = 1000\frac{J}{kgK}$, $L = 4.0 \times 10^5 \frac{J}{kg}$, $\rho = 2400\frac{kg}{m^3}$ and $\kappa = 100\frac{W}{mK}$. The liquidus temperature, at an initial solute concentration of 5.0 % is, $T_l = 904.2\ K$. The eutectic temperature is, $T_{eut} = 821.2\ K$, and the melting temperature of the pure solvent is $T_f = 921.2\ K$. The slope of the liquidus line is $m_l = 3.4$, and the partition coefficient is $k = 0.15$. The initial conditions for temperature is $905.2\ K$. At time equal to zero, the left boundary is set to a temperature of $621.2\ K$, and solidification proceeds.

Tables 4 and 5 present results for different time step sizes, with the Scheil local scale model, for $N = 50$ and $N = 200$, respectively. The results in both tables are with first order time differencing. Again, we see all of the same performance trends observed in the pure material Stefan problem. Adding the complexity of the mushy zone to the temperature function, and the associated local nonlinear solves within this function, has not effected the algorithm performance in a significant way. In Table 5 very large time steps were used as compared to the explicit stability limit. This does not appear to have had

a significant effect in terms of accuracy or computational effort.

It should be noted that for $N = 200$ and and a time step of 10 the eutectic front is moving approximately one grid cell per time step on average. This statement is arrived at knowing that 56 time steps were taken and that the eutectic front has moved out past $X = 0.1$, which is 50 finite volumes from the left boundary. Since this eutectic front has a characteristic velocity, V_{front}, we refer to this simulation with $N = 200$ and time step of 10 as having a front CFL of 1, i.e. $(V_{front}\Delta t)\ /\ \Delta x = 1$. The final entry in Table 5 is asymptotically running at a front CFL of 5, which means that the solidification front is moving an average of 5 grid cells per time step.

Figure 7 displays the nonlinear convergence history at two different time steps for $N = 50$ and $\Delta t = 20.0$. During time step 31 the front does not cross into a new cell, while in time step 28 the front does move into a new cell.

Figure 8 is a plot of solid volume fraction, ϵ_s, for three different solutions, isolated around the front at time = 500, with $N = 50$. Here it is seen that ϵ_s profiles exhibit very little difference between a $\Delta t = 20.0$ simulation and a $\Delta t = 1.0$ simulation. This is significant since our method requires a factor of 5.3 less linear solves (to reach time = 500) for a time step of 20. More importantly, when a temporally second-order method (Crank-Nicolson) is used for the $\Delta t = 20.0$ simulation there is virtually no difference in the solution when compared to the $\Delta t = 1.0$ simulation. The second order method requires the same number of linear solves to reach a time of 500, but the L_2 error of the second order method is 1.6×10^{-3} while the L_2 error of the first order method is 8.0×10^{-3}. Also, a front CFL of 0.5 is observed

for this problem with $\Delta t = 20.0$. Our method can integrate with these large time steps while maintaining accuracy since the front is naturally captured with our enthalpy-based energy equation.

Finally, Table 6 presents results using the lever rule as the local scale model on a grid of $N = 50$. In comparing Tables 4 and 6 we see very little difference in the performance of the method with a different local scale models.

6 Conclusions

A new nonlinear iterative algorithm has been put forth for phase change problems. The method possesses Newton-like super-linear convergence characteristics with out forming the Jacobian from Newton's method. Enthalpy is the dependent variable and phase change fronts evolve with out fixing front positions. Our method conserves energy globally and locally for any size time step. The performance of the method has been demonstrated on a variety of pure material isothermal solidification problems and on the nonisothermal solidification of a binary eutectic alloy with two simple local scale models. Our method has demonstrated the ability to retain a high degree of accuracy and efficiency even when solidification fronts are moving multiple grid cells per time step.

7 Acknowledgments

This work was supported under the auspices of the U.S. Department of Energy under DOE contract W-7405-ENG-36 at Los Alamos National Laboratory, and as part of the Accelerated Strategic Computing Initiative (ASCI).

References

[1] V.R. Voller. An Overview of Numerical Methods for Solving Phase Change Problems. *Advances in Numerical Heat Transfer*, 1:341–375, 1996.

[2] C.R. Swaminathan and V.R. Voller. Towards a general numerical scheme for solidification systems. *Int. J. Heat Mass Transfer*, 40:2859, 1996.

[3] V.R. Voller, C.R. Swaminathan, and B.G. Thomas. Fixed Grid Techniques for Phase Change Problems: A Review. *Int. J. Num. Meth. Eng.*, 30:875, 1990.

[4] P.R. McHugh and D.A. Knoll. Comparison of Standard and Matrix-Free Implementations of Several Newton-Krylov Solvers. *AIAA J.*, 32(12):2394–2400, 1994.

[5] D.A. Knoll, P.R. McHugh, and D.E. Keyes. Newton-Krylov methods for low Mach number compressible combustion. *AIAA J.*, 34(5):961–967, 1996.

[6] R.W. Johnson, P.R. McHugh, and D.A. Knoll. High-order scheme implementation using Newton-Krylov solution methods. *Numerical Heat Transfer, Part B*, 31:295–312, 1997.

[7] C.T. Kelly and J. Rulla. Solution of the time discretized Stefan problem by Newton's method. *Nonlinear Analysis, Theory Methods and Applications*, 14:851, 1990.

[8] V. Alexiades and A.D. Solomon. *Mathematical Modeling of Melting and Freezing Processes*. Hemisphere, Washington, 1993.

[9] T.W. Clyne and W. Kurz. Solute redistribution during solidification with rapid state diffusion. *Metallurgical Transactions A*, 12:965, 1981.

[10] C.Y. Wang and C. Beckermann. A multiphase solute diffusion model for dendritic solidification. *Metallurgical Transactions A*, 24:2787, 1993.

[11] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.

[12] C.T. Kelly. *Iterative Methods for Linear and Nonlinear Equations*. SIAM Frontiers in Applied Mathematics, Philidelphia, 1995.

[13] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, 11:450–481, 1990.

[14] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving non-symetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856, 1986.

Table 1: Algorithm performance as a function of time step size on Stefan problem, upto time = 8.0, $St = 0.1$, $X_f^{base} = 1.05$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position (X_f)	L_2 Error
2, Δt=1.0e-2	806	1.9	1533	1.05	6.0e-4
10, Δt=5.0e-2	166	1.27	210	1.05	3.5e-3
20, Δt=1.0e-1	86	1.68	145	1.05	6.2e-3
40, Δt=2.0e-1	46	2.26	104	1.05	1.2e-2
80, Δt=4.0e-1	26	3.54	94	1.05	2.2e-2

Table 2: Algorithm performance as a function of time step size on the Stefan problem, upto time = 4.0, $St = 1.0$, $X_f^{base} = 1.54$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position (X_f)	L_2 Error
1, Δt=5.0e-3	806	1.9	1550	1.54	2.6e-3
2, Δt=1.0e-2	406	1.6	658	1.54	5.6e-3
4, Δt=2.0e-2	206	1.3	261	1.53	1.2e-2
10, Δt=5.0e-2	86	1.57	135	1.52	2.8e-2
20, Δt=1.0e-1	46	2.0	92	1.49	5.8e-2
40, Δt=2.0e-1	26	2.5	66	1.46	9.6e-2

Table 3: Algorithm performance as a function of time step size on the Stefan problem, upto time = 2.0, $St = 10.0$ $X_f^{base} = 1.31$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position (X_f)	L_2 Error
1, Δt=5.0e-3	406	1.68	684	1.31	2.5e-3
2, Δt=1.0e-2	206	1.19	245	1.31	5.4e-3
4, Δt=2.0e-2	106	1.36	145	1.31	1.1e-2
10, Δt=5.0e-2	46	1.78	82	1.30	2.8e-2
20, Δt=1.0e-1	26	2.5	65	1.29	5.4e-2

Table 4: Algorithm performance as a function of time step size on the Voller binary eutectic problem, Scheil model, upto time = 500.0, $N = 50$, $\Delta t_{exp} = 0.768$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position $\epsilon_s = 0.5$	L_2 Error
0.32, Δt=0.25	2006	1.02	2046	0.157	
1.3, Δt=1.0	506	1.4	708	0.157	4.0e-4
6.5, Δt=5.0	106	2.55	270	0.1569	2.0e-3
13.0, Δt=10.0	56	3.19	179	0.1567	4.1e-3
26.0, Δt=20.0	31	4.32	134	0.1565	8.3e-3
130.0, Δt=100.0	11	6.3	70	0.1564	2.9e-2

Table 5: Algorithm performance as a function of time step size on the Voller binary eutectic problem, Scheil model, upto time = 500.0, $N = 200$, $\Delta t_{exp} = 0.048$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position $\epsilon_s = 0.5$	L_2 Error
20.83, Δt=1.0	506	2.22	1123	0.1572	6.4e-4
104.17, Δt=5.0	106	4.0	424	0.1571	3.2e-3
208.33, Δt=10.0	56	5.95	333	0.1570	6.4e-3
416.67, Δt=20.0	31	7.35	228	0.1567	1.27e-2
1041.67, Δt=50.0	16	8.68	139	0.1565	2.68e-2

Table 6: Algorithm performance as a function of time step size on the Voller binary eutectic problem, lever model, upto time = 500.0, $N = 50$, $\Delta t_{exp} = 0.768$

Explicit Fac. and Time step	Number of time steps	Number of Newton Its. per time step	Total Linear Solves	Front Position $\epsilon_s = 0.5$	L_2 Error
0.32, Δt=0.25	2006	1.02	2046	0.155	8.6e-5
1.3, Δt=1.0	506	1.33	673	0.155	3.7e-4
6.5, Δt=5.0	106	2.3	243	0.155	1.9e-3
13.0, Δt=10.0	56	2.6	146	0.155	3.7e-3
26.0, Δt=20.0	31	3.6	112	0.155	7.6e-3

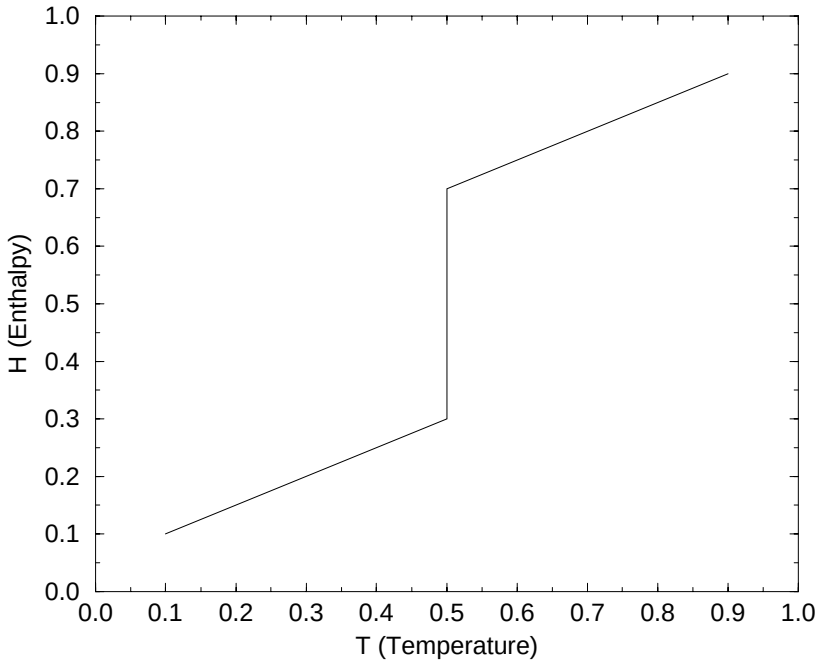

Figure 1: Pure material $H(T)$ relation, $T_m = 0.5$ and $L = 0.4$

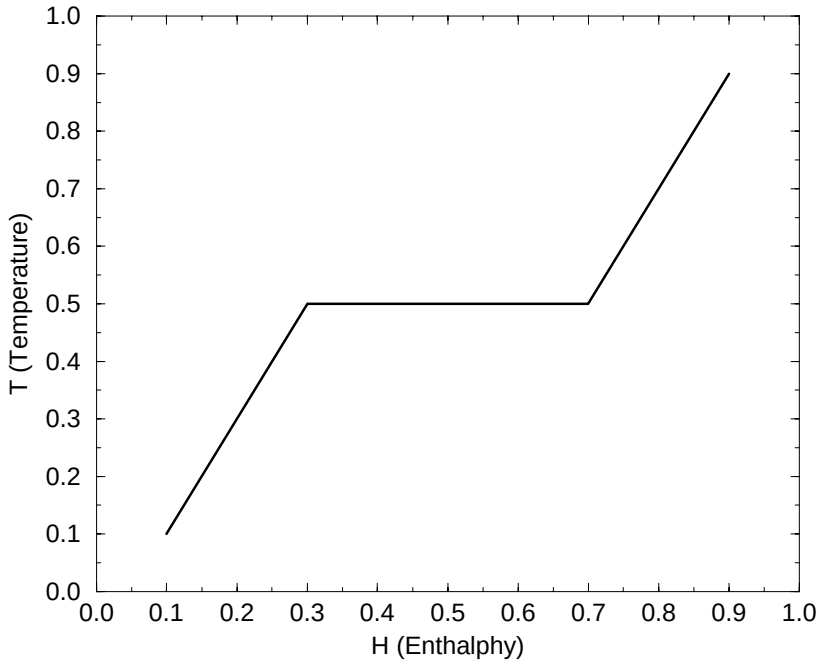

Figure 2: Pure material $T(H)$ relation, $T_m = 0.5$ and $L = 0.4$

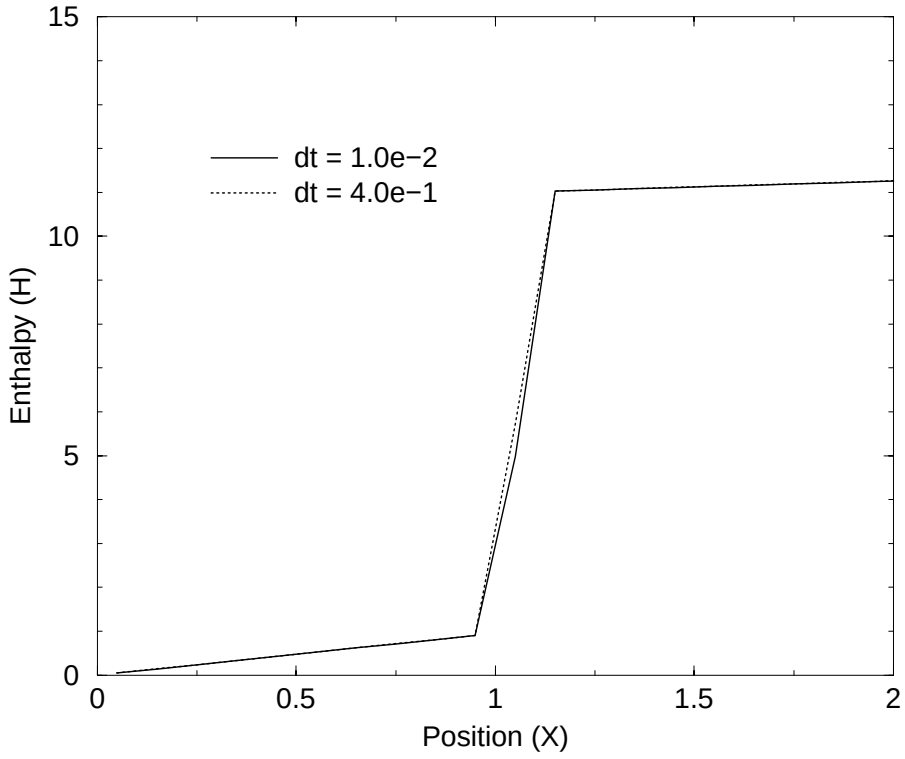

Figure 3: Enthalpy profiles at time = 8.0 for two different time step sizes ($St = 0.1$).

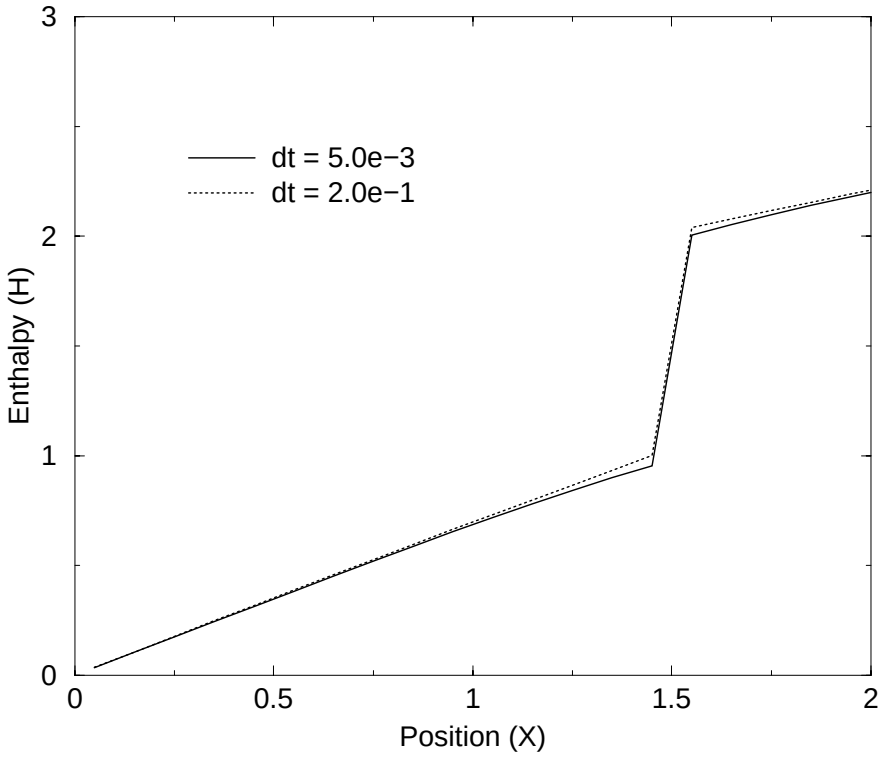

Figure 4: Enthalpy profiles at time = 4.0 for two different time step sizes ($St = 1.0$).

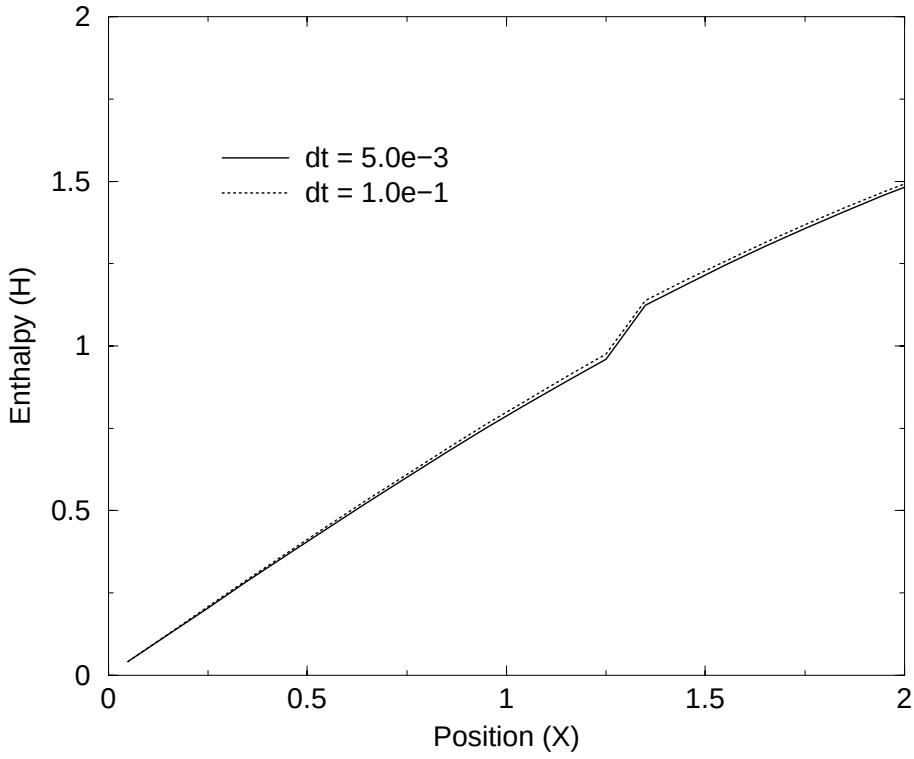

Figure 5: Enthalpy profiles at time = 2.0 for two different time step sizes ($St = 10.0$).

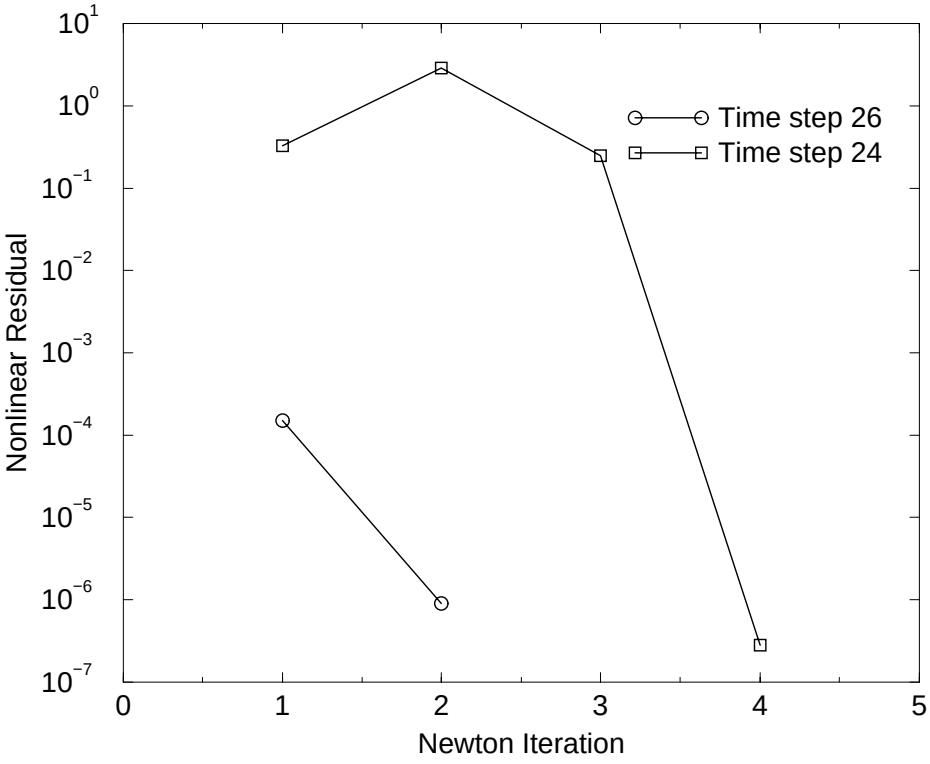

Figure 6: Nonlinear convergence for $St = 0.1$, $\Delta t = 0.4$, time step numbers 24 and 26.

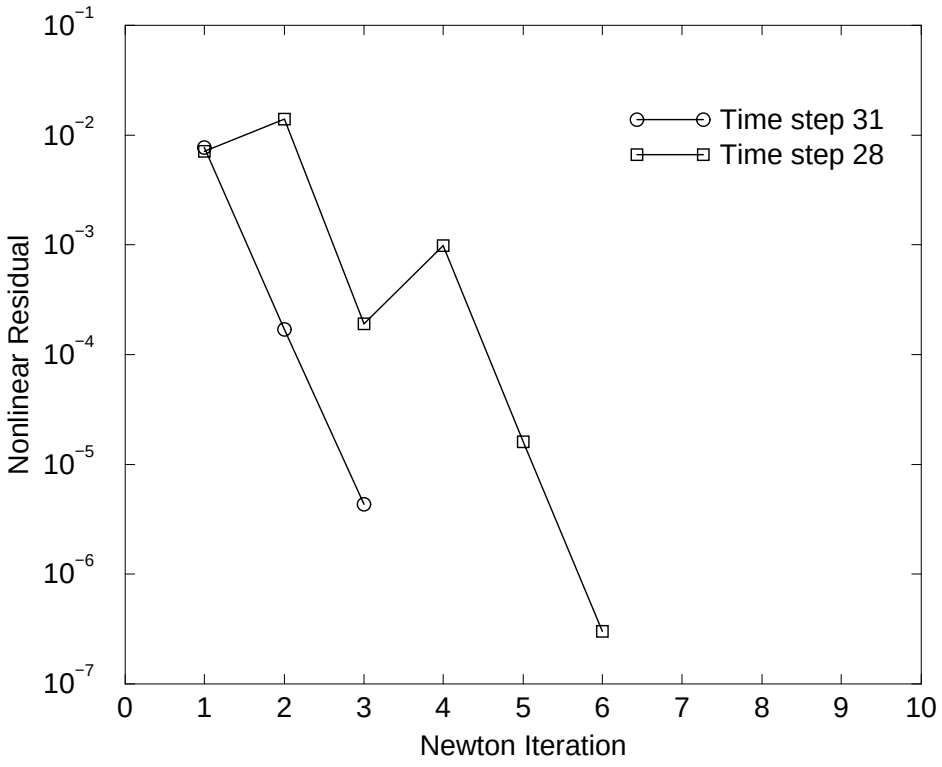

Figure 7: Nonlinear convergence for binary eutectic model problem, Scheil local scale model, $N = 50$, $\Delta t = 20.0$, time step numbers 28 and 31.

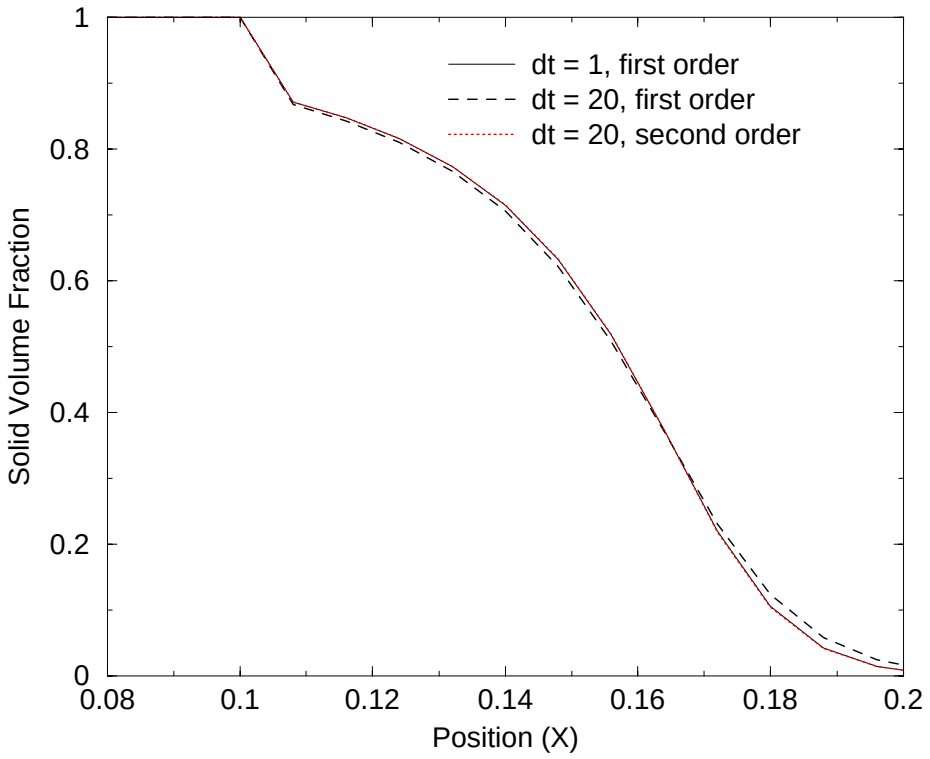

Figure 8: Comparison of solid volume fraction solution for binary eutectic model problem, Scheil local scale model, $N = 50$, $\Delta t = 1.0$ and 20.0.